

시스템 프로그래밍

Assignment1-1

Class : A

Professor : 김태석 교수님

Student ID : 2022202065

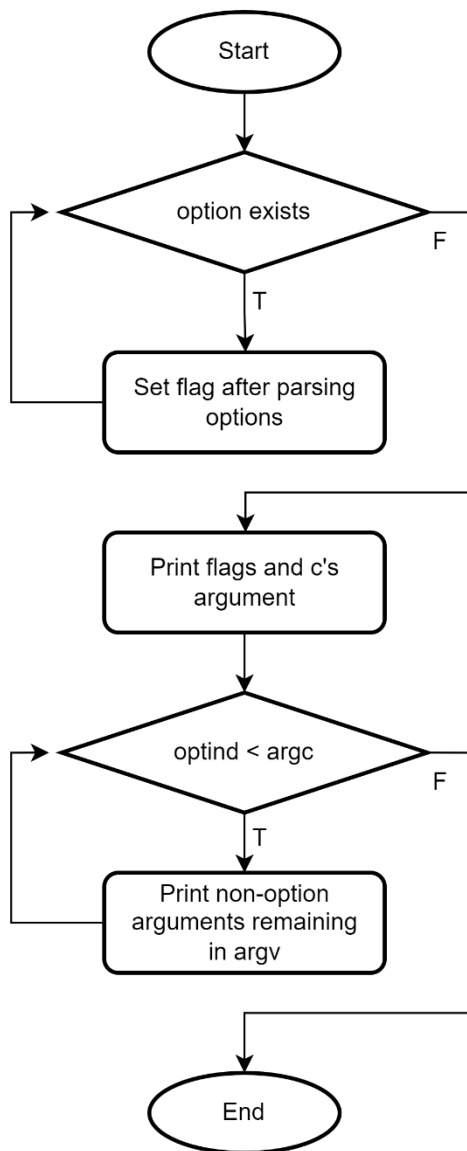
Name : 박나림

Introduction

FTP (File Transfer Protocol)를 구현하는 프로젝트의 첫번째 단계로, FTP Commands 를 옵션, 인수별로 각각 파싱하여 받기 위해 문자열을 조작하는 것이 목표이다.

이를 위해 프로그램 실행 옵션을 파싱하기 위한 함수인 'getopt()'를 사용해 보도록 한다. 옵션은 '-'로 시작하며, '-'이 뒤에 붙는 옵션은 추가 인수가 필요하다. 옵션이 없으면 -1 을 리턴하며 분석을 완료한다. 이러한 특성을 이용하여 c 파일과 Makefile 을 생성하고 여러 테스트케이스로 진행해보도록 한다.

Flow chart



Pseudo code

```
while (until there are options)
```

```
{
```

```
    switch (returnValue)
```

```
    {
```

```
        set flag for each option;
```

```
    }
```

```
}
```

```
Print flags and c(option)'s argument;
```

```
while (non-option argument < argv)
```

```
{
```

```
    Print non-option arguments remaining in argv;
```

```
}
```

결과화면

-kw2022202065_opt.c

```
///////////////////////////////////////////////////////////////////
// File Name      : kw2022202065_opt.c
// Date           : 2024/03/30
// OS              : Ubuntu 20.04.6 LTS 64bits
// Author          : Park Na Rim
// Student ID     : 2022202065
// -----
// Title  : System Programming Assignment #1-1 (ftp server)
// Description : Parameter parsing (-a, -b, -c [str] by using getopt().
/////////////////////////////////////////////////////////////////

#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv)
{
    int aflag = 0, bflag = 0;          // Flags for options -a and -b
    char *cvalue = NULL;               // Arguments for option -c
    int c = 0;                         // Return value of getopt()
    opterr = 0;                       // Error variable

    ///Repeat until there are options on the command line===//
    while((c = getopt(argc, argv, "abc:")) != -1)
    {
        switch (c)
        {
            case 'a':
                aflag++;
                break;
            case 'b':
                bflag++;
                break;
            case 'c':
                cvalue = optarg; // Save Option's argument value
                break;
            case '?':
                // Unknown option character
                break;
        }
    }

    ///=====Print flags and c's argument=====//
    printf("aflag = %d, bflag = %d, cvalue = %s\n", aflag, bflag, cvalue);

    ///=====Print non-option arguments remaining in argv=====//
    while(optind < argc)
    {
        printf("Non-option argument %s\n", argv[optind++]);
    }

    return 0;
}
```

옵션이 없을 때까지 getopt()로 파싱하여 각 옵션별로 switch 문을 이용하여 flag 를 증가시킨다. c 옵션의 경우, 추가 인수가 더 들어오므로 optarg 를 cvalue 에 저장시킨다. 그 후 각 flag 와 argument 를 출력한다음 argv 에 남아있는 non-option 에 대해 optind 로 출력시킨다.

-Makefile

```
kw2022202065_opt: kw2022202065_opt.c
    gcc -o kw2022202065_opt kw2022202065_opt.c

~
~
```

-Result

```
kw2022202065@ubuntu:~$ ./kw2022202065_opt
aflag = 0, bflag = 0, cvalue = (null)
kw2022202065@ubuntu:~$ ./kw2022202065_opt -a -b
aflag = 1, bflag = 1, cvalue = (null)
kw2022202065@ubuntu:~$ ./kw2022202065_opt -ab
aflag = 1, bflag = 1, cvalue = (null)
kw2022202065@ubuntu:~$ ./kw2022202065_opt -c foo
aflag = 0, bflag = 0, cvalue = foo
kw2022202065@ubuntu:~$ ./kw2022202065_opt -cfoo
aflag = 0, bflag = 0, cvalue = foo
kw2022202065@ubuntu:~$ ./kw2022202065_opt arg1
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2022202065@ubuntu:~$ ./kw2022202065_opt -a arg1
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2022202065@ubuntu:~$ ./kw2022202065_opt -c foo arg1
aflag = 0, bflag = 0, cvalue = foo
Non-option argument arg1
kw2022202065@ubuntu:~$ ./kw2022202065_opt -a -
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -
kw2022202065@ubuntu:~$ ./kw2022202065_opt -aa
aflag = 2, bflag = 0, cvalue = (null)
kw2022202065@ubuntu:~$ ./kw2022202065_opt -d -a
aflag = 1, bflag = 0, cvalue = (null)
kw2022202065@ubuntu:~$ ./kw2022202065_opt foo foo
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument foo
Non-option argument foo
```

순서대로 각 테스트 케이스를 실행한 결과이다. 지정된 a, b, c 옵션이 들어오면 flag 를 증가시키고 인수값을 출력한다. 옵션이 아닌 인수가 들어오면 다음 줄에 해당 인수를 출력한다. -d 와 같이 지정되지 않은 옵션의 경우엔 그냥 넘어간다. 단순히 -만 입력했을 때는 non-option argument 로써 출력된다.

고찰

이번에 getopt() 함수를 써보면서 사용법을 자세히 익힐 수 있었다. 기존에 알고 있었던 문자열 조작함수인 strcpy, strcat 등을 이용하면 null 이나 index, 포인터들에 대해서는 따로 직접 구현을 해주는 식으로 사용해야 됐었는데, getopt()같은 경우엔 이미 구현된 optarg, optind, opterr, optopt 가 있어서 더 편하다고 느껴졌다.

또한 아직 리눅스가 익숙하지 않았었는데, 이번에 테스트 케이스들을 진행해보면서 옵션 쓰는 법을 공부할 수 있었다. 옵션은 -로 시작하며 지정된 옵션일 경우 띄어쓰기를 안 해도 된다는 것을 알게 되었다. 또한 :뒤에 오는 것은 추가 인수가 필요하다는 것도 새롭게 알게 되었다. 모든 비옵션들은 뒤에 있기 때문에 argv 에 남아있어서 optind 를 이용하여 출력하는 것도 새로웠다.

전체적으로 c 파일과 Makefile 을 vi 로 쓰고 터미널에서 실행시키는 방식도 더 익숙해질 수 있었던 것 같다.

Reference

강의자료 참고